

# Generalized Majority Decision Reducts

Sebastian Widz\* and Sebastian Stawicki†

\*Systems Research Institute, Polish Academy of Sciences  
 ul. Newelska 6, 01-447 Warsaw, Poland

†Institute of Mathematics, University of Warsaw  
 ul. Banacha 2, 02-097 Warsaw, Poland

sebastian.widz@gmail.com, sebastian.stawicki@gmail.com

**Abstract**—We discuss several new methods for constructing approximate decision reducts from the rough set theory. We introduce generalized approximate majority decision reducts, which are an extension to standard approximate decision reducts known from literature but with improved calculation performance and complexity. We also discuss the relationship and differences of the new approximate decision reduct notion with so called decision bireducts – another type of approximate decision reducts.

**Keywords**—Feature Subset Selection; Rough Sets; Approximate Decision Reducts; Decision Bireducts

## I. INTRODUCTION

ATTRIBUTE subset selection plays an important role in knowledge discovery [1]. It establishes the basis for more efficient classification, prediction and approximation models. It also provides the users with a better insight into data dependencies. In this paper, we concentrate on attribute subset selection methods originating from the theory of rough sets [2]. There are numerous rough-set-based algorithms aimed at searching for so called *reducts* – irreducible subsets of attributes that satisfy predefined criteria for keeping enough information about decisions. Those criteria are verified on the training data and, usually, they encode the risk of misclassification by if-then decision rules with their antecedents referring to the values of investigated attribute subsets and their consequents referring to decisions.

Original definition of a reduct is quite restrictive, requiring that it should determine decisions or, if data inconsistencies do not allow full determinism, provide the same level of information about decisions as the complete set of attributes. There are a number of approaches to formulate and search for approximate or inexact reducts, which *almost* preserve the decision information [3]. Approximate reducts are usually smaller than standard ones, providing the basis for learning more efficient classifiers [4], [5].

In our previous work [6] we compared approximate decision reducts based on feature subset quality functions [7], [8] with decision bireducts [9] – another extension to the rough set framework of decision reducts. In short, a decision bireduct is an irreducible feature subset which preserves the information about the decision but only on selected objects from the data set. The feature subset cannot be reduced as well as the object subset cannot be further expanded without violating the decision functional dependence.

In this article we show new methods for calculating approximate decision reducts based on generalized majority decision (GMD) function. The concept of GMD function might be considered as an modification to a very well known concept of generalized decision function known from the rough set theory. We also consider a few variations of decision rule-based classifiers induction from the discussed approximate decision reducts. We introduce the concept of *exception* decision rules which are created during the reduct calculation process. The use of *exceptions* can be helpful in case we need to construct simplified decision models [10] and preserve more information about the original data set. We also explain the relationship of the proposed methods to the notion of decision bireducts and propose a new algorithm for decision bireducts calculation based on GMD function.

The remainder of this paper is organized as follows. Section II briefly describes the concept of generalized decision function. In Section III we explain approximate decision reducts. In Sections IV and V we present a new definition of approximate decision reducts and show new algorithms in Section VI. In Section VII we introduce the concept of *exception* decision rules. In Section VIII we show relationship to the decision bireducts and a new way of decision bireduct calculation. Finally, we show experiment results in Section IX and conclude our work in Section X.

## II. DECISION REDUCTS AND GENERALIZED DECISION

We use the standard notion of a decision system to represent data [2]. By a decision system we mean a tuple  $\mathbb{A} = (U, A \cup \{d\})$ , where  $U$  is a set of objects and  $A$  is a set of attributes and  $d \notin A$  is a distinguished decision attribute. For simplicity, we refer to the elements of  $U$  using their ordinal numbers  $i = 1, \dots, |U|$ , where  $|U|$  denotes the cardinality of  $U$ . We treat attributes  $a \in A$  as functions  $a : U \rightarrow V_a$ ,  $V_a$  denoting  $a$ 's value domain. The values  $v_d \in V_d$  correspond to decision classes that we want to describe using the values of attributes in  $A$ . Each subset  $B \subseteq A$  partitions the space  $U$  into *equivalence classes*. For such a division we get the partition space denoted as  $U/B = \{E_1, \dots, E_t\}$  where  $E_t \subseteq U$ . Each equivalence class is defined as  $E_t = \{x \in U : B(x) = v_t\}$  where  $v_t$  is a vector of values on attributes  $B$ . We will further refer the set of objects with particular decision  $k$  as  $X_k$ .

**Definition 1.** We say that  $B \subseteq A$  is a decision reduct for  $\mathbb{A} = (U, A \cup \{d\})$  if and only if it is an irreducible subset

of features such that each pair  $i, j \in U$  satisfying inequality  $d(i) \neq d(j)$  is discerned by  $B$ .

Definition 1 works well in case of consistent decision tables, however in case of inconsistent data no attribute reduction is possible, in fact, even the whole attribute set  $A$  cannot be considered to be a decision reduct itself. There are many alternative definitions of decision reducts that could be applied to inconsistent decision tables, e.g., subsets of features that preserve the same positive region as the full set of attributes. Another example could be a decision reduct based on generalized decision function [2]:

**Definition 2.** For a given decision table  $\mathbb{A} = (U, A \cup \{d\})$ , we say that a **generalized decision function** is a function  $\partial_d : 2^U \rightarrow 2^{V_d}$  defined as follows:

$$\partial_d(E) = \{k : X_k \cap E \neq \emptyset\} \quad (1)$$

The cardinality of a generalized decision may be used to express the level of inconsistency in describing decision attribute by subsets of features. In particular, if  $|\partial_d([x]_A)| = 1$ , for any  $x \in U$ , then the decision table  $\mathbb{A} = (U, A \cup \{d\})$  is said to be *consistent*. Otherwise it is *inconsistent*. The example of a generalized decision is presented in Table I in column denoted as  $\partial_d([x]_A)$ .

**Definition 3.** Let  $\mathbb{A} = (U, A \cup \{d\})$  be given. We say that  $B \subseteq A$  is a  $\partial$ -decision superreduct if and only if the following condition holds:

$$\bigvee_{x \in U} \partial_d([x]_B) = \partial_d([x]_A) \quad (2)$$

We say that  $B$  is a  $\partial$ -decision reduct if and only if it is a  $\partial$ -superreduct and none of its proper subsets satisfy the above condition.

### III. FOUNDATIONS OF APPROXIMATE REDUCTS

There are a variety of methods of searching for approximate decision reducts (e.g. [11], [12], [13]). The criteria usually include formulas for functions measuring degrees of decision information induced by subsets of features and thresholds for those functions' values specifying which subsets of attributes are *good enough*. The choice of functions may depend on the nature of particular data sets and methods of learning classifiers based on reduced sets of attributes. In order to follow the filter approach to feature subset selection, we need to design some measures that evaluate particular feature subsets in the selection process. From this point of view, the rough set literature may be regarded as a source of measures that draw correspondence between feature subsets and rule-based classifiers corresponding to those subsets, where each subset of attributes  $B \subseteq A$  yields a set of decision rules based on all combinations of its values in  $U$  and the consecutive value is chosen according to some criteria.

Let us define a measure  $F : P(A) \rightarrow \mathfrak{R}$  which evaluates the degree of influence  $F(B)$  of subset  $B \subseteq A$  in  $d$ . Then one can decide which features may be removed from  $A$  without significant loss of accuracy.

**Definition 4.** Let  $\mathbb{A} = (U, A \cup \{d\})$  and approximation threshold  $\varepsilon \in [0, 1)$  be given. We say that  $B \subseteq A$  is an  $(F, \varepsilon)$ -approximate decision reduct if and only if it satisfies the following condition:

$$F(B) \geq (1 - \varepsilon)F(A) \quad (3)$$

and none of its proper subsets  $C \subseteq B$  does it.

There are many examples how function  $F$  can be defined. Let us focus on two examples. The first is *Majority* measure (further denoted as  $M$ ), proposed in [7]:

$$M(B) = \sum_{E \in U/B} \frac{|E|}{|U|} \max_{k \in V_d} \frac{|X_k \cap E|}{|E|} \quad (4)$$

The function  $M$  points to a decision value that appears the most frequently within a particular equivalence class  $E$ . It means that if we need to decide which decision should be attached to objects from a particular class, then we always choose the most frequent decision that was observed in the training data. Based on the measure  $M$ , one could generate decision rules for which the rules' right sides are the most frequent decisions for  $E \in U/B$ . Another example of  $F$  refers to a Bayesian extension of the classical rough set model, where rules induced by a given subset of attributes are pointing at the decision classes which become maximally frequent comparing to their overall occurrence in the data. This function called *Relative information gain* measure (further denoted as  $R$ ) was proposed in [8].

$$R(B) = \frac{1}{|V_d|} \sum_{E \in U/B} \max_{k \in V_d} \frac{|X_k \cap E|}{|X_k|} \quad (5)$$

In our research the permutation based *REDORD* algorithm introduced in [14] is used as a baseline algorithm (see Algorithm 1).

---

#### Algorithm 1 Permutation-based $(F, \varepsilon)$ -REDORD Algorithm

---

**Input:**  $\mathbb{A} = (U, A \cup \{d\})$ ,  $\varepsilon \in [0, 1)$ ,

$\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ ,  $n = |A|$

**Output:**  $(F, \varepsilon)$  approximate decision reduct  $B \subseteq A$

```

1:  $B \leftarrow A$ 
2: for  $i = 1 \rightarrow n$  do
3:   if  $F(B \setminus \{a_{\sigma(i)}\}) \geq (1 - \varepsilon)F(A)$  then
4:      $B \leftarrow B \setminus \{a_{\sigma(i)}\}$ 
5:   end if
6: end for
7: return  $B$ 

```

---

### IV. GENERALIZED MAJORITY DECISION REDUCTS

Approximate decision reducts calculation methods allow some level of interaction with users designing decision models. However, this interaction is only limited to a single parameter describing the degree of approximation that refers to the allowed overall misclassification rate of a decision model. What is more restricting is that the decision model based on

approximate decision reduct is always pointing to majority decision within particular equivalence classes. We would like to give the user the ability to decide about the decision classes distribution within each  $E$ . Based on the idea of generalized decision we formulate *generalized majority decision* (described in this section) and *generalized approximation majority decision* concepts. The approximate version is described in the next section.

**Definition 5.** For a given decision table  $\mathbb{A} = (U, A \cup \{d\})$ , we say that a **generalized majority decision function** is a function  $m_d : 2^U \rightarrow 2^{V_d}$  defined as follows:

$$m_d(E) = \{k : |X_k \cap E| = \max_j (|X_j \cap E|)\} \quad (6)$$

Generalized majority decision function reflects the choice of most frequent decision classes within subsets  $E \subseteq U$ . In Table I we present an example of generalized decision and generalized majority decision attributes denoted  $\partial_d([x]_A)$  and  $m_d([x]_A)$  respectively.

Table I  
DECISION TABLE AND GENERALIZED DECISION AND GENERALIZED MAJORITY DECISION ATTRIBUTES

$Id$	$a_1$	$a_2$	$a_3$	$a_4$	$d$	$\partial_d([x]_A)$	$m_d([x]_A)$
$x_1$	1	1	2	2	0	{0,1}	{0,1}
$x_2$	1	1	2	2	1	{0,1}	{0,1}
$x_3$	1	1	2	2	1	{0,1}	{0,1}
$x_4$	1	1	2	2	0	{0,1}	{0,1}
$x_5$	3	3	1	2	1	{0,1}	{0,1}
$x_6$	3	3	1	2	0	{0,1}	{0,1}
$x_7$	2	3	1	2	1	{1}	{1}
$x_8$	1	2	2	1	2	{0,1,2}	{1,2}
$x_9$	1	2	2	1	2	{0,1,2}	{1,2}
$x_{10}$	1	2	2	1	1	{0,1,2}	{1,2}
$x_{11}$	1	2	2	1	1	{0,1,2}	{1,2}
$x_{12}$	1	2	2	1	0	{0,1,2}	{1,2}
$x_{13}$	2	1	1	1	1	{1}	{1}
$x_{14}$	2	2	1	1	0	{0}	{0}

In case of consistent decision tables  $|m_d([x]_A)| = 1$ . In case of inconsistent decision tables the cardinality of  $m_d([x]_A)$  will be greater than one only in a case of an equivalence class with more than one equally distributed decision values. However, let us notice that one could be interested in calculating  $m_d([x]_B)$  for any subset of features  $B \subseteq A$  containing the smaller number of attributes. In such a case the equivalence classes are relatively large, and equal frequencies of decision values are more common.

**Definition 6.** Let  $\mathbb{A} = (U, A \cup \{d\})$  be given. We say that  $B \subseteq A$  is an  $(m, =)$ -decision superreduct if and only if the following condition holds:

$$\bigvee_{x \in U} m_d([x]_B) = m_d([x]_A) \quad (7)$$

We say that  $B$  is an  $(m, =)$ -decision reduct if and only if it is an  $(m, =)$ -superreduct and none of its proper subsets satisfy the above condition.

**Proposition 1.** Let  $\mathbb{A} = (U, A \cup \{d\})$  be given. If  $B \subseteq A$  is an  $(m, =)$ -decision superreduct, then:

$$\forall x, y \in U m_d([x]_A) \neq m_d([y]_A) \Rightarrow \exists a \in B a(x) \neq a(y) \quad (8)$$

*Proof:* First let us notice that we can transform the above equation to the following form:

$$\forall x, y \left( \forall a \in B a(x) = a(y) \Rightarrow m_d([x]_A) = m_d([y]_A) \right) \quad (9)$$

Let us consider any  $x, y$  such that  $\forall a \in B a(x) = a(y)$ . We need to show that  $m_d([x]_A) = m_d([y]_A)$ . Based on Definition (6) we know that  $m_d([x]_A) = m_d([x]_B)$  and that  $m_d([y]_A) = m_d([y]_B)$ . Because  $[x]_B = [y]_B$ , then  $m_d([x]_B) = m_d([y]_B)$ , this also  $m_d([x]_A) = m_d([y]_A)$ . Let us take any  $x \in U$ . We need to show that  $m_d([x]_B) = m_d([x]_A)$ . Let us take any  $v_k \in m_d([x]_A)$ . We need to show that  $v_k \in m_d([x]_B)$ . If  $v_k \in m_d([x]_A)$ , then based on Equation (9) we have  $v_k \in m_d([y]_A)$  for all  $y \in [x]_B$ . This means, that  $v_k$  has maximal frequency among all decision classes for all  $A$ -indiscernibility classes contained in  $[x]_B$ . This means that it also has maximum frequency in  $[x]_B$ . Thus:

$$m_d([x]_B) \subseteq m_d([x]_A) \quad (10)$$

always holds.

In Proposition (3) we will show that if the intersection of  $m_d([y]_A)$  sets for all  $y \in [x]_B$  for a given  $x \in U$  is not empty, then the following holds:

$$m_d([x]_B) \subseteq m_d([x]_A) \quad (11)$$

Based on Equation (9) we know that this intersection is nonempty. Based on Equations (9) and (10) we have

$$m_d([x]_B) = m_d([x]_A) \quad (12)$$

■

Let us propose also another method of using the  $m_d$  function to define attribute reduction criteria, where we are not interested in the full discernibility with regard to the symbolic representation of generalized majority decision but we apply a weaker condition.

**Definition 7.** Let  $\mathbb{A} = (U, A \cup \{d\})$  be given. We say that  $B \subseteq A$  is an  $(m, \subseteq)$ -decision superreduct if and only if the following condition holds:

$$\bigvee_{x \in U} m_d([x]_B) \subseteq m_d([x]_A) \quad (13)$$

We say that  $B$  is an  $(m, \subseteq)$ -decision reduct if and only if it is an  $(m, \subseteq)$ -superreduct and none of its proper subsets satisfy the above condition.

The following result emphasizes the importance of Definition 7.

**Proposition 2.** Let  $\mathbb{A} = (U, A \cup \{d\})$  be given.  $B \subseteq A$  is an  $(m, \subseteq)$ -decision superreduct if and only if the following condition holds:

$$M(B) = M(A) \quad (14)$$

Let us further notice that Definition 7 is more flexible than Definition 6 and it usually allows more attribute reduction.

**Proposition 3.** Let  $\mathbb{A} = (U, A \cup \{d\})$  be given.  $B \subseteq A$  is an  $(m, \subseteq)$ -decision superreduct if and only if:

$$\bigvee_{x \in U} \bigcap_{y \in [x]_B} m_d([y]_A) \neq \emptyset \quad (15)$$

*Proof:* Let us take any  $x \in U$  and let us consider all objects  $y \in [x]_B$ . We know that:

$$\bigvee_{y \in [x]_B} m_d([y]_B) \subseteq m_d([y]_A) \quad (16)$$

Because  $y \in [x]_B$ , then  $[y]_B = [x]_B$ . Thus:

$$\bigvee_{y \in [x]_B} m_d([x]_B) \subseteq m_d([y]_A) \quad (17)$$

and therefore:

$$\bigvee_{y \in [x]_B} m_d([x]_B) \subseteq \bigcap_{y \in [x]_B} m_d([y]_A) \quad (18)$$

Thus,  $\bigcap_{y \in [x]_B} m_d([y]_A)$  cannot be empty.

Let us take any  $x \in U$  and let us consider all objects  $y \in [x]_B$ . Because, we know that:

$$\bigcap_{y \in [x]_B} m_d([y]_A) \neq \emptyset \quad (19)$$

we also know that there exists such  $v$  that  $v \in m_d([y]_A)$  for every  $y \in [x]_B$ . By Definition (5) we know that  $v$  must be one of the most frequent decision values on all equivalence classes  $[y]_A$  such that  $y \in [x]_B$ . This is because  $[x]_B$  is a sum of some disjoint blocks  $E \in U/A$ , which are represented by  $[y]_A$ . For that reason it must be also one of the most frequent values in the whole  $[x]_B$  i.e.  $v \in m_d([x]_B)$ . For any other decision  $v'$  contained in  $m_d([x]_B)$ , it would have to be as frequent in  $[x]_B$  as the decision value  $v$ . But if would require for  $v'$  to be as frequent as  $v$  in all blocks  $E_A \in U/A, E_A \subseteq E_B$ . Thus, it would be also  $v' \in m_d(E_A)$  for all such blocks. In particular we would have:

$$m_d([x]_B) \subseteq m_d([x]_A) \quad (20)$$

**Proposition 4.** Let  $\mathbb{A} = (U, A \cup \{d\})$  be given.  $B \subseteq A$  is an  $(m, \subseteq)$ -decision superreduct if and only if:

$$\bigvee_{x \in U} \bigcap_{y \in [x]_B} m_d([y]_A) = m_d([x]_B) \quad (21)$$

Let us illustrate the difference between  $(m, \subseteq)$ - and  $(m, =)$ -decision reducts. Let us consider a decision table presented in Table II. All three objects have exactly the same values on feature subset  $B$  but different values on attribute  $a$ . According to definition of  $(m, =)$ -decision reduct attribute  $a$  cannot be removed.  $(m, =)$ -decision reduct requires that generalized majority decision sets are exactly the same for merged equivalence classes. In case of  $(m, \subseteq)$ -decision reduct this attribute can be removed as it requires only set inclusion.

Table II  
ATTRIBUTE REMOVAL IN  $(m, \subseteq)$ - AND  $(m, =)$ -decision reducts.

$Id$	$B$	$a$	$m_d([x]_{B \cup \{a\}})$
$x_1$	...	1	{0,1}
$x_2$		2	{0,1,2}
$x_3$		3	{0,2}

## V. GENERALIZED APPROXIMATE MAJORITY DECISION

The generalized majority decision function points only to a single most frequent decision for a given equivalence class. Its drawback is that it does not keep any information about minority decision classes and there is no way to control the threshold how much information about the minority decision classes should be maintained. We propose to determine this threshold based on decision class distribution within equivalence classes and a level of approximation given by the user.

Let us introduce a definition of *generalized approximate majority decision* function. In comparison to Equation 1 we include not only the most frequent decision values but also those which are almost as frequent as the majority decision values. We express this statement by introducing a threshold parameter  $\varepsilon$  that controls the ratio between allowed decision values and the majority decision.

**Definition 8.** For a given decision table  $\mathbb{A} = (U, A \cup \{d\})$  and approximation threshold  $\varepsilon \in [0, 1)$ , we say that a **generalized approximate majority decision function** is a function  $m_d^\varepsilon : 2^U \rightarrow 2^{V_a}$  defined as follows:

$$m_d^\varepsilon(E) = \{k : |X_k \cap E| \geq (1 - \varepsilon) \max_j |X_j \cap E|\} \quad (22)$$

**Proposition 5.** For  $\varepsilon = 0$  and  $\varepsilon \rightarrow 1^-$  we have:

$$\bigvee_E m_d^0(E) = m_d(E) \quad \text{and} \quad \bigvee_E \lim_{\varepsilon \rightarrow 1^-} m_d^\varepsilon(E) = \partial_d(E)$$

Let us propose an alternative way of the attribute reduction based on the new concept of  $(m^\varepsilon, \cap)$ -decision reduct and generalized decision set intersections. Let us further notice that this approach is equivalent to the previous one for  $\varepsilon = 0$ .

**Definition 9.** Let  $\mathbb{A} = (U, A \cup \{d\})$  be given. We say that  $B \subseteq A$  is an  $(m^\varepsilon, \cap)$ -decision superreduct if and only if:

$$\bigvee_{x \in U} \bigcap_{y \in [x]_B} m_d^\varepsilon([y]_A) \neq \emptyset \quad (23)$$

We say that  $B$  is an  $(m^\varepsilon, \cap)$ -decision reduct if and only if it is an  $(m^\varepsilon, \cap)$ -decision superreduct and none of its proper subsets satisfy the above condition.

**Proposition 6.** Let  $\mathbb{A} = (U, A \cup \{d\})$  be given. The following holds for every subset  $B \subseteq A$ :

$$\bigvee_{x \in U} \bigcap_{y \in [x]_B} m_d^\varepsilon([y]_A) \subseteq m_d^\varepsilon([x]_B) \quad (24)$$

**Proposition 7.** If  $B \subseteq A$  is an  $(m^\varepsilon, \cap)$ -decision superreduct then the following inequality holds:

$$M(B) \geq (1 - \varepsilon)M(A) \quad (25)$$

Let us notice that in case of Proposition 2 we have the equivalence between of  $B \subseteq A$  being an  $(m, \subseteq)$ -decision reduct and given the equality  $M(B) = M(A)$ , whereas in case of Proposition 7 and  $(m^\varepsilon, \cap)$ -decision reduct we have only the implication.

In the traditional decision rule induction approach one can use contents of a decision table and the set of attributes  $B$  to induce decision rules. Similarly, decision rules calculated with respect to the generalized majority decisions take such a form that the ascendants are based on values of  $B$  over objects  $x \in U$  but consequents point at disjunctions of possible decisions belonging to  $m_d^\varepsilon(E)$ . For example, let us consider objects  $\{x_8, x_9, x_{10}, x_{11}, x_{12}\}$  from Table I. Based on these objects we can produce the following decision rule:

$$a_1 = 1 \wedge a_2 = 1 \wedge a_3 = 2 \wedge a_4 = 1 \Rightarrow d = 1 \vee d = 2$$

It is important to note that such a rule could be rewritten as two separate rules, both having the same support and confidence:

$$a_1 = 1 \wedge a_2 = 2 \wedge a_3 = 2 \wedge a_4 = 1 \Rightarrow d = 1$$

$$a_1 = 1 \wedge a_2 = 2 \wedge a_3 = 2 \wedge a_4 = 1 \Rightarrow d = 2$$

In fact all decision values belonging to generalized approximate majority decision set are treated as equally distributed within a particular equivalence class. Moreover, all decision rules based on this equivalence class are considered to have equal quality support and confidence.

## VI. HEURISTIC SEARCH FOR APPROXIMATE REDUCTS

The above results could be utilized in a procedure for attribute reduction. First we calculate generalized majority decision for the whole attribute set  $A$ . Next we can remove the attribute  $a$  whenever the Equality 23 is satisfied. This means that if the intersection of all generalized decision of smaller equivalence classes induced by  $B \cup \{a\}$  is nonempty we can remove attribute  $a$  without a significant loss of information. Algorithm 2 presents the procedure that utilizes *Reduce* function described in Algorithm 3. In Table III and Table IV we give brief examples of attribute removal tries, the unsuccessful and successful, respectively.

Table III  
IMPOSSIBLE REMOVAL OF ATTRIBUTE  $a_2$

$Id$	$a_1$	$a_3$	$a_4$	$m_d^\varepsilon([x]_B)$	
$\{x_1, \dots, x_4\}$	1	2	2	$\{0,1\}$	✓
$\{x_5, x_6\}$	3	1	2	$\{0,1\}$	✓
$\{x_7\}$	2	1	2	$\{1\}$	✓
$\{x_8 \dots x_{12}\}$	1	2	1	$\{1,2\}$	✓
$\{x_{13} \cup x_{14}\}$	2	1	1	$\{1\} \cap \{0\} = \emptyset$	✗

Let us focus on decision rules that could be generated before and after removal of attribute  $a_1$  based on objects  $\{x_5, x_6, x_7\}$ . Before removal of attribute  $a_1$  that rules are as follow:

$$a_1 = 3 \wedge a_2 = 3 \wedge a_3 = 1 \wedge a_4 = 2 \Rightarrow d = 0 \vee d = 1$$

$$a_1 = 2 \wedge a_2 = 3 \wedge a_3 = 1 \wedge a_4 = 2 \Rightarrow d = 1$$

---

### Algorithm 2 Generalized Majority Decision Reduct (GMDR2)

---

**Input:**  $\mathbb{A} = (U, A \cup \{d\})$ ,  $\varepsilon \in [0, 1)$ ,  $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ ,  $n = |A|$ ,

$\mathbb{A}_{temp}$  - temporary decision table for storing equivalence classes)

**Output:**  $B \subseteq A$

```

1: Calculate Generalized Majority Decision  $m_d^\varepsilon(E_A)$  for all
   objects in  $\mathbb{A}$ 
2:  $B \leftarrow A$ 
3:  $E_B \leftarrow \text{CreateEquivalenceClasses}(A)$ 
4: for  $i = 1 \rightarrow n$  do
5:    $E_C \leftarrow \text{Reduce}(E_B, B, a_{\sigma(i)})$ 
6:   if  $E_C \neq E_B$  then
7:      $E_B = E_C$ 
8:      $B \leftarrow B \setminus \{a_{\sigma(i)}\}$ 
9:   end if
10: end for
11: return  $B$ 

```

---



---

### Algorithm 3 Attribute reduction function

---

**Input:** Collection of equivalence classes  $E_B \in U/B$ ,

Attribute subset  $B \subseteq A$ , Candidate attribute  $a \in B$  for removal

**Output:** Collection of equivalence classes  $E_C \in U/C$  where  $C \subseteq B$  if attribute  $a \in B$  was removed or  $E_B \in U/C$  otherwise

```

1: function REDUCE( $E_B, B, \{a\}$ )
2:    $C \leftarrow B \setminus \{a\}$ 
3:    $E_C \leftarrow \emptyset$ 
4:   for all EquivalenceClasses  $e \in E_B$  do
5:      $DEC \leftarrow \emptyset$ 
6:      $v_B \leftarrow \text{GetInstance}(e)$ 
7:      $v_C \leftarrow \text{Remove}(v_B, \{a\})$ 
8:      $e_{tmp} \leftarrow \text{Find}(E_C, v_C)$ 
9:     if  $e_{tmp} \neq \text{NULL}$  then
10:       $DEC \leftarrow \text{GetDec}(e_{tmp}) \cap \text{GetDec}(e)$ 
11:      if  $|DEC| > 0$  then
12:         $\text{SetDec}(e_{tmp}, DEC)$ 
13:      else
14:        return  $E_B$ 
15:      end if
16:    else
17:       $\text{AddEquivalenceClass}(E_C, e_{tmp})$ 
18:    end if
19:  end for
20:  return  $E_C$ 
21: end function

```

---

After removal of attribute  $a_1$ :

$$a_2 = 3 \wedge a_3 = 1 \wedge a_4 = 2 \Rightarrow d = 1$$

From this perspective we can understand the described procedure as creating a lowe approximation of decision rules generated on the whole attribute set.

Table IV  
SUCCESSFUL REMOVAL OF ATTRIBUTE  $a_1$

$Id$	$a_2$	$a_3$	$a_4$	$m_d^\varepsilon([x]_B)$	
$\{x_1, \dots, x_4\}$	1	2	2	$\{0, 1\}$	✓
$\{x_5, x_6\} \cup \{x_7\}$	3	1	2	$\{0, 1\} \cap \{1\} = \{1\}$	✓
$\{x_8 \dots x_{12}\}$	2	2	1	$\{1, 2\}$	✓
$\{x_{13}\}$	1	1	1	$\{1\}$	✓
$\{x_{14}\}$	2	1	1	$\{0\}$	✓

## VII. APPROXIMATE DECISION REDUCTS WITH EXCEPTIONS

Let us propose an another way of expressing levels of allowed inconsistency in the decision model. We introduce a new approximation threshold  $\phi$  that relates to the maximal ratio of objects that can be misclassified. By analogy, we introduce a new definition of  $(m^\phi, \cap)$ -decision reduct which utilizes the generalized majority decision function.

**Definition 10.** Let  $\phi \in [0, 1]$  and  $\mathbb{A} = (U, A \cup \{d\})$  be given. Subset  $B \subseteq A$  is an  $(m^\phi, \cap)$ -decision superreduct if and only if there exists subset  $X \subseteq U$  satisfying inequality  $|X| \geq (1 - \phi)|U|$ , such that the following condition holds:

$$\bigvee_{x \in X} \bigcap_{y \in [x]_B \cap X} m_d([y]_A) \neq \emptyset \quad (26)$$

We say that  $B$  is an  $(m^\phi, \cap)$ -decision reduct if and only if it is an  $(m^\phi, \cap)$ -decision superreduct and none of its proper subsets satisfy the above conditions.

The idea behind  $(m^\phi, \cap)$ -decision reducts  $B \subseteq A$  is to cover sufficiently the data set with decision rules constructed using attributes in  $B$  and provide a comparable level of information to decision rules constructed using all features from  $A$ . If  $B \subseteq A$  is an  $(m^\phi, \cap)$ -decision superreduct, then there exists subset  $X \subseteq U$  satisfying inequality  $|X| \geq (1 - \phi)|U|$  and condition (26) such that  $X$  is definable by  $A$ , which means that  $X$  is the set-theoretic sum of some equivalence classes of  $U/A$ . This explains in what sense subsets  $X \subseteq U$  can be treated as set-theoretic sum of supports of premises of rules induced by  $A$  whose information is kept at comparable level after shortening them to more general decision rules induced by  $B$ . Thus, from now on, we will implicitly assume that all subsets  $X \subseteq U$  discussed in context of  $(m^\phi, \cap)$ -decision reducts take a form of sums of some equivalence classes of  $U/A$ .

**Proposition 8.** If  $B \subseteq A$  is an  $(m^\phi, \cap)$ -decision superreduct, then the following inequality holds

$$M(B) \geq M(A) - \phi \quad (27)$$

Let us notice that we could combine definitions of  $(m^\phi, \cap)$ -decision reduct with  $(m^\varepsilon, \cap)$ -decision reduct and formulate a combined definition of  $(m^{\varepsilon, \phi}, \cap)$ -decision reduct. For simplicity, we will further assume that  $\varepsilon = 0$ .

In Section IV we discussed an example showing that the attribute  $a_2$  cannot be removed. This was due to the fact that objects  $x_{13}$  and  $x_{14}$  have an empty intersection of their generalized majority decision sets. In other words, there was

### Algorithm 4 Attribute reduction function with exceptions

**Input:** Collection of equivalence classes  $E_B \in U/B$ , Approximation degree  $\phi \in [0, 1]$ ; Attribute subset  $B \subseteq A$ , Candidate attribute  $a \in B$  for removal

**Output:** Updated exception rules set  $R_{ex}$ , Collection of equivalence classes  $E_C \in U/C$  where  $C \subseteq B$  if attribute  $a \in B$  was removed or  $E_B \in U/B$  otherwise

```

1: function REDUCE2( $E_B, B, \{a\}, \phi, R$ )
2:    $C \leftarrow B \setminus \{a\}$ 
3:    $E_C \leftarrow \emptyset$ 
4:    $w_C \leftarrow \text{GetWeight}(E_B)$ 
5:   for all EquivalenceClasses  $e \in E_B$  do
6:      $DEC \leftarrow \emptyset$ 
7:      $v_B \leftarrow \text{GetInstance}(e)$ 
8:      $v_C \leftarrow \text{Remove}(v_B, \{a\})$ 
9:      $e_{tmp} \leftarrow \text{Find}(E_C, v_C)$ 
10:    if  $e_{tmp} \neq \text{NULL}$  then
11:       $DEC \leftarrow \text{GetDec}(e_{tmp}) \cap \text{GetDec}(e)$ 
12:      if  $|DEC| > 0$  then
13:         $\text{SetDec}(e_{tmp}, DEC)$ 
14:      else
15:         $w_C \leftarrow w_C - |e|$ 
16:        if  $w_C \leq (1 - \phi)|U|$  then
17:          return  $E_B$ 
18:        end if
19:         $\text{StoreExceptionRule}(R, e_{tmp})$ 
20:      end if
21:    else
22:       $\text{AddEquivalenceClass}(E_C, e)$ 
23:    end if
24:  end for
25:   $\text{SetWeight}(E_C, w_C)$ 
26:  return  $E_C$ 
27: end function

```

only one object that was blocking the algorithm from reducing the number of features. If we removed either object  $x_{13}$  or object  $x_{14}$  from the discussed decision table the reduction would be possible but we would also lose some information. Exception rules allow us to reduce the number of features and at the same time to save additional information about lost objects. Let us focus on Tables V and VI where we again consider the same example but this time we can remove attribute  $a_2$ . We create the following exception rule (supported only by one object  $\{x_{14}\}$ ) but with confidence = 1.0)

$$a_1 = 2 \wedge a_2 = 1 \wedge a_3 = 1 \wedge a_4 = 1 \Rightarrow d = 0$$

In the next step we try to remove another attribute, again if some objects do not allow attribute removal we can remove them and save in form of exception rules, remembering that no more than  $\phi \cdot |U|$  objects can be removed in total. This procedure is presented as Algorithm 4.

During the classification phase, the exception rules are always searched in the first place. If a proper exception rule is

Table V  
SUCCESSFUL REMOVAL OF ATTRIBUTE  $a_2$

$Id$	$a_1$	$a_3$	$a_4$	$m_d^{\sigma}([x]_B)$	
$\{x_1, \dots, x_4\}$	1	2	2	$\{0,1\}$	✓
$\{x_5, x_6\}$	3	1	2	$\{0,1\}$	✓
$\{x_7\}$	2	1	2	$\{1\}$	✓
$\{x_8 \dots x_{12}\}$	1	2	1	$\{1,2\}$	✓
$\{x_{13} \cup x_{14}\}$	2	1	1	$\{1\} \cap \{0\} = \emptyset$	✓(-1)

Table VI  
STATE AFTER REMOVAL OF ATTRIBUTE  $a_2$  – OBJECT  $x_{14}$  WAS REMOVED

$Id$	$a_1$	$a_3$	$a_4$	$m_d^{\sigma}([x]_B)$
$\{x_1, \dots, x_4\}$	1	2	2	$\{0,1\}$
$\{x_5, x_6\}$	3	1	2	$\{0,1\}$
$\{x_7\}$	2	1	2	$\{1\}$
$\{x_8 \dots x_{12}\}$	1	2	1	$\{1,2\}$
$\{x_{13}\}$	2	1	1	$\{1\}$

found, we do not search any other rules related to this reduct, i.e., any shorter more general rule that had been created in the subsequent stages of the learning process.

### VIII. BIREDUCTS AND GENERALIZED MAJORITY DECISION

The first formulation of decision bireducts occurred in [9], [6], where their Boolean characterization was described and a simple ordering algorithm aimed at their heuristic extraction from data was proposed by an analogy to classical decision reducts. The motivation to introduce decision bireducts was to allow for explicit analysis whether classifiers that use different selected subsets of attributes do not repeat classification mistakes on the same objects in the training data set. Experiments reported in [9] show that diversification of subsets of attributes with this respect may be important in practice.

**Definition 11.** Let  $\mathbb{A} = (U, A \cup \{d\})$  and subsets  $B \subseteq A$ ,  $X \subseteq U$  be given. We say that  $B$  determines  $d$  within  $X$ , further denoted as  $B \Rightarrow_X d$ , if and only if  $B$  discerns all pairs  $u_i, u_j \in X$  such that  $d(u_i) \neq d(u_j)$ . Further, we say that the pair  $(X, B)$  is a decision bireduct if and only if the following holds:

- 1) There is  $B \Rightarrow_X d$ ,
- 2) There is no proper subset  $B' \subsetneq B$  such that  $B' \Rightarrow_X d$ ,
- 3) There is no proper superset  $X' \supsetneq X$  such that  $B \Rightarrow_{X'} d$ .

The original ordering algorithm to calculate a decision bireduct for a given data set is presented in Algorithm 5. It is recalled here as a reference point to a modified version of the algorithm (presented as Algorithm 6) that works with equivalence classes instead of single objects and with generalized majority decision intersections constraint instead of the original discernibility condition.

A major difference of the ordering algorithm for decision bireducts extraction, comparing to the ordering algorithms obtaining decision reducts (reduction on attributes only, e.g., Algorithm 2), can be observed in the main loop iterating on both the attributes and the objects at the same time. An appropriate action is taken, i.e., either removal of unnecessary attribute or addition of an object to the result, on base

#### Algorithm 5 Original Ordering Decision Bireduct Algorithm

**Input:** a decision table  $\mathbb{A} = (U, A \cup \{d\})$ ,  $\sigma$  – a permutation of a set  $\{1, \dots, |U| + |A|\}$

**Output:** a decision bireduct for  $\mathbb{A}$

```

1:  $X_0 \leftarrow \emptyset, B_0 \leftarrow A$ 
2: for  $i = 1 \rightarrow |U| + |A|$  do
3:    $B_i \leftarrow B_{i-1}$ 
4:    $X_i \leftarrow X_{i-1}$ 
5:   if  $\sigma(i) \leq |U|$  then
6:     if  $B_i \Rightarrow_{X_i \cup \{u_{\sigma(i)}\}} d$  then
7:        $X_i \leftarrow X_i \cup \{u_{\sigma(i)}\}$ 
8:     end if
9:   else
10:    if  $B_i \setminus \{a_{\sigma(i)-|U|}\} \Rightarrow_{X_i} d$  then
11:       $B_i \leftarrow B_i \setminus \{a_{\sigma(i)-|U|}\}$ 
12:    end if
13:  end if
14: end for
15: return  $(X_{|U|+|A|}, B_{|U|+|A|})$ 

```

#### Algorithm 6 Ordering Generalized Majority Decision Bireduct Algorithm

**Input:** a decision table  $\mathbb{A}^{m_d} = (U^{m_d}, A \cup \{m_d\})$  obtained from  $\mathbb{A} = (U, A \cup \{d\})$ , where equivalence classes induced by  $A$  are considered objects in  $\mathbb{A}^{m_d}$ , i.e.,  $U^{m_d} = U/A$  and where  $m_d$  is used to compute their decision values,  $\sigma$  – a permutation of a set  $\{1, \dots, |U^{m_d}| + |A|\}$

**Output:** a decision bireduct for  $\mathbb{A}^{m_d}$

```

1:  $X_0 \leftarrow \emptyset, B_0 \leftarrow A$ 
2: for  $i = 1 \rightarrow |U^{m_d}| + |A|$  do
3:    $B_i \leftarrow B_{i-1}$ 
4:    $X_i \leftarrow X_{i-1}$ 
5:   if  $\sigma(i) \leq |U^{m_d}|$  then
6:     if  $Test^{m_d}(X_i \cup \{u_{\sigma(i)}^{m_d}\}, B_i)$  then
7:        $X_i \leftarrow X_i \cup \{u_{\sigma(i)}^{m_d}\}$ 
8:     end if
9:   else
10:    if  $Test^{m_d}(X_i, B_i \setminus \{a_{\sigma(i)-|U^{m_d}|}\})$  then
11:       $B_i \leftarrow B_i \setminus \{a_{\sigma(i)-|U^{m_d}|}\}$ 
12:    end if
13:  end if
14: end for
15: return  $(X_{|U^{m_d}|+|A|}, B_{|U^{m_d}|+|A|})$ 

```

of a given condition for the generalized majority decisions intersections.

It is worth to notice that the reduct and bireduct versions of the generalized majority decision algorithms are not equivalent, i.e., there exist data sets for which certain results can be obtained only by one of the methods.

As an example, let us consider the following fragment of a decision table. All presented objects have the same values on feature subset  $B$  and the specified values on attributes  $a$  and

**Algorithm 7** Test function used by Algorithm 6.

---

```

1: function  $Test^{ma}(X \subseteq U^{ma}, B \subseteq A)$ 
2:    $\mathcal{X}_{Test} \leftarrow \emptyset$ 
3:   for all EquivalenceClasses  $e \in X$  do
4:      $v \leftarrow \text{Remove}(\text{GetInstance}(e), A \setminus B)$ 
5:      $e_{tmp} \leftarrow \text{Find}(\mathcal{X}_{Test}, v)$ 
6:     if  $e_{tmp} \neq \text{NULL}$  then
7:        $DEC \leftarrow \text{GetDec}(e_{tmp}) \cap \text{GetDec}(e)$ 
8:       if  $|DEC| > 0$  then
9:          $\text{SetDec}(e_{tmp}, DEC)$ 
10:      else
11:        return false
12:      end if
13:    else
14:       $\text{AddEquivalenceClass}(\mathcal{X}_{Test}, e)$ 
15:    end if
16:  end for
17:  return true
18: end function

```

---

b.

$B$	$a$	$b$	$m_d([x]_{B \cup \{a,b\}})$
	0	0	{0, 1, 2}
	0	1	{0, 1, 3}
...	1	0	{0, 2, 3}
	1	1	{1, 2, 3}

Let us consider the generalized majority decision reduct computation flow. If we try to remove only one column  $a$  or  $b$ , respectively, we would get the following:

$B$	$b$	$m_d([x]_{B \cup \{b\}})$	$B$	$a$	$m_d([x]_{B \cup \{a\}})$
...	0	{0, 2}	...	0	{0, 1}
	1	{1, 3}		1	{2, 3}

We can observe that in either case the second column cannot be removed because  $\{0, 2\} \cap \{1, 3\}$  as well as  $\{0, 1\} \cap \{2, 3\}$  are equal to  $\emptyset$ . Let us now consider a fragment of the input permutation  $a, 1, 2, 3, b, 4$  to the generalized majority decision bireduct algorithm. We can remove the attribute  $a$ , add equivalence classes 1, 2, 3 and remove the attribute  $b$  because the intersection of the appropriate decisions are equal to  $\{0\}$ :

$B$	$m_d([x]_B)$
	{0, 1, 2}
...	{0, 1, 3}
	{0, 2, 3}
...	{1, 2, 3}

The subsequent addition of the equivalence class 4 is not possible due to the fact that intersection of the decision  $\{1, 2, 3\}$  with all other already added to the decision bireduct is equal to  $\emptyset$ . Therefore, we would finish with a decision bireduct consisting of the attribute subset  $B$  and the subset containing only the first three equivalence classes.

## IX. EXPERIMENTS

We conducted our experiments on a collection of benchmark data sets available from the University of California at Irvine (UCI) Repository [15]. We present our results based on two data sets (*DNA-splices* and *ZOO*) but similar results can be observed on other benchmark data. We compared different types of approximate decision reducts calculation: the approximate decision reducts (ADR) calculated according to Algorithm 1 and three variations on the generalized approximate majority decision reducts (GMDA) with exceptions rules (EXEP) and with exceptions pointing to unknown decision (GAPS) and without any exception rules (NONE) but with the same reduction criteria. We have also tested each algorithm based on  $M$  and  $R$  measures. In case of GMDA procedure, the indications  $M$  and  $R$  correspond to the way how generalized approximate majority decision set was constructed. In case of the  $R$  measure we considered the relative frequency of a decision class within each equivalence class prior to a decision frequency in the testing set.

The data sets that were not given with *a priori* split into training and testing data sets, have been tested using 5-fold cross validation. Every experiment was conducted at least 20 times. We followed the well known ensemble technique to construct a classifier consisting of 10 reducts, each yielding a single weak classifier with a set of decision rules (including exceptions) calculated according to a particular method. We calculated 200 decision reducts and selected reducts with the smallest number of features. In case of exception rules we also included exceptions in calculation of the average reduct size but we used weighted average, where number of recognized objects is understood as a weight of the reduct.

Like in any ensemble-based classifier, some combination technique of classifier outputs must be applied. We used voting based on decision rule confidence, but other rule measures can be also utilized – see our previous research [16] where we described and tested several methods for rule identification and voting mechanisms.

## X. CONCLUSIONS

We have presented several new methods for calculating approximate decision reducts. All of presented methods are based on the generalized majority decision function and the concept of analyzing intersections between generalized majority decision sets. We consider new methods to be more flexible in expressing the user preferences to control the distribution of decision classes inside each equivalence class. The presented methods are also less computationally expensive in comparison to the standard methods for approximate decision reducts, as they allow, in most situations, a quicker test for attribute removal. We have also presented the idea of constructing a decision rule-based classifier that use a special type of rules referred as exceptions. These rules are produced during the decision model construction and allow constructing models with a reduced dimensionality but with information about original training data preserved. We have experimentally validated our propositions and algorithms. Results show that

Figure 1. Average accuracy on *DNA-splices* data set

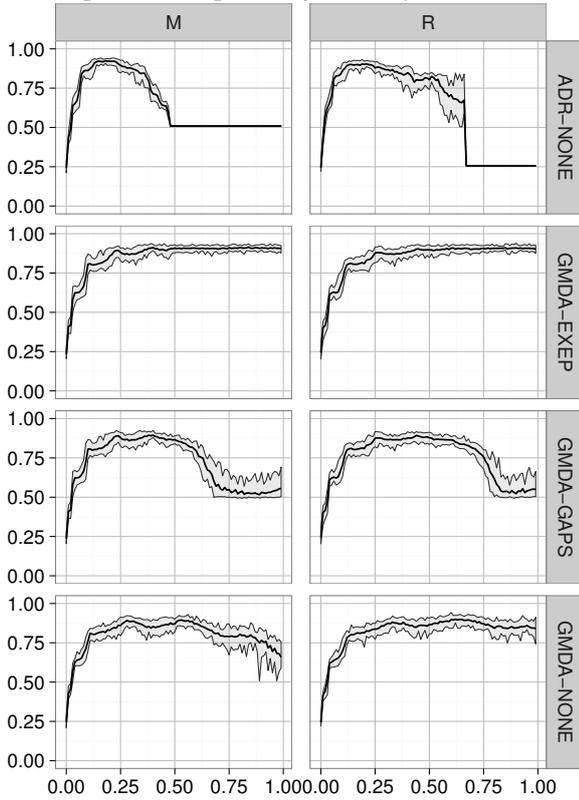


Figure 3. Average accuracy on *ZOO* data set

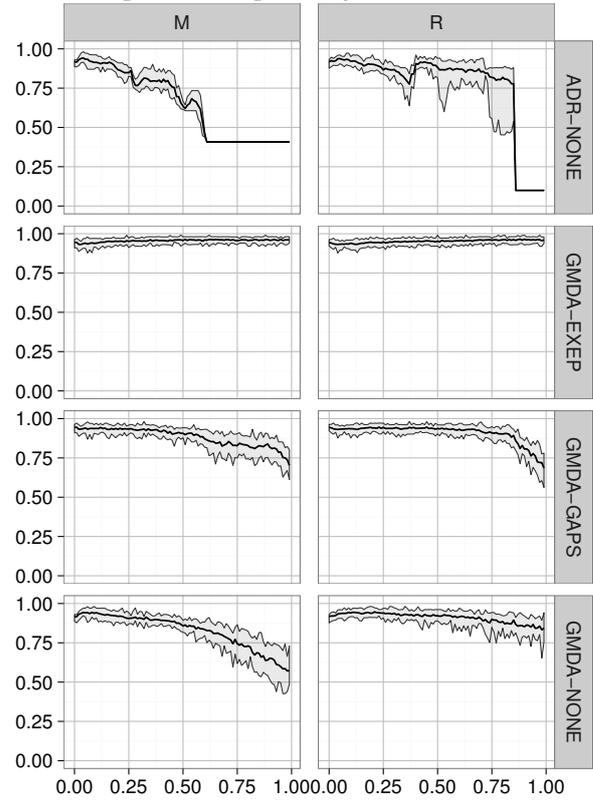


Figure 2. Average length of decision rules – *DNA-splices* data set

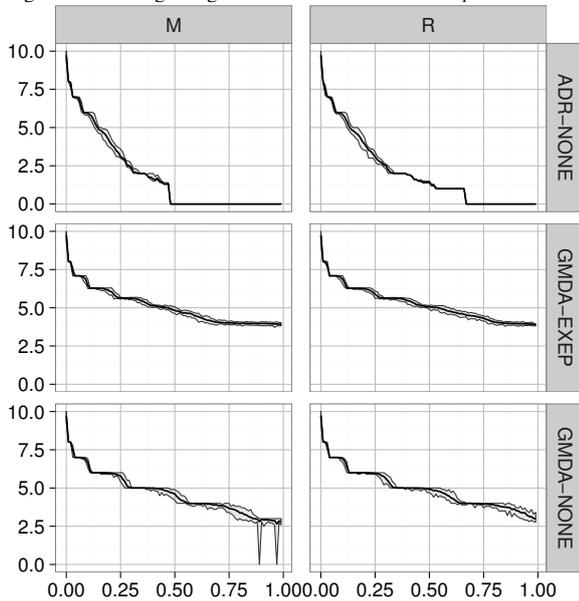
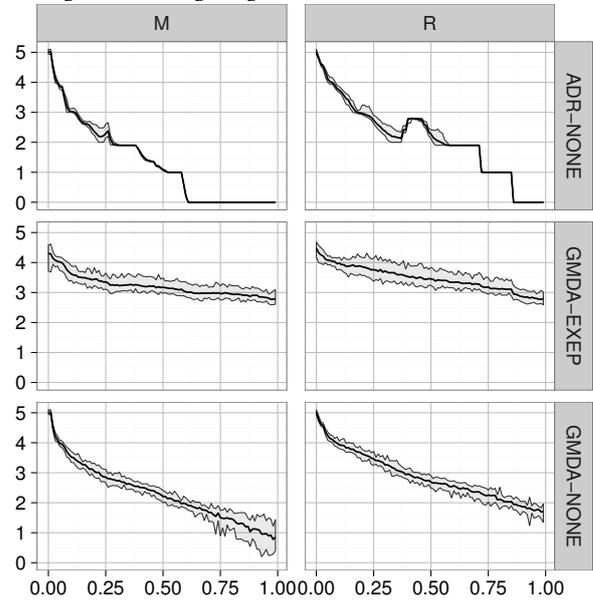


Figure 4. Average length of decision rules – *ZOO* data set



the new way of constructing approximation decision reducts have very comparable accuracy but is less computationally complex. Last but not least, we compared our method with bireduct framework and proposed new method for bireduct calculation based on the generalized majority decision set intersections. We plan to analyze the last method more deeply

in the nearest future.

REFERENCES

[1] H. Liu and H. Motoda, Eds., *Computational Methods of Feature Selection*. Chapman & Hall/CRC, 2008.  
 [2] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information sciences*, vol. 177, no. 1, pp. 3–27, 2007.

- [3] D. Ślęzak, "Rough Sets and Functional Dependencies in Data: Foundations of Association Reducts," *LNCS Transactions on Computational Science*, vol. V, pp. 182–205, 2009.
- [4] S. Widz and D. Ślęzak, "Approximation Degrees in Decision Reduct-based MRI Segmentation," in *FBIT*. IEEE, 2007, pp. 431–436.
- [5] A. Janusz and S. Stawicki, "Applications of Approximate Reducts to the Feature Selection Problem," in *RSKT*. Springer, 2011, pp. 45–50.
- [6] S. Stawicki and S. Widz, "Decision bireducts and approximate decision reducts: Comparison of two approaches to attribute subset ensemble construction," in *Proc. of FedCSIS'2012 Conf.* IEEE, 2012, pp. 331–338.
- [7] D. Ślęzak, "Normalized Decision Functions and Measures for Inconsistent Decision Tables Analysis," *Fundamenta Informaticae*, vol. 44, no. 3, pp. 291–319, 2000.
- [8] D. Ślęzak and W. Ziarko, "The Investigation of the Bayesian Rough Set Model," *International Journal of Approximate Reasoning*, vol. 40, no. 1-2, pp. 81–91, 2005.
- [9] D. Ślęzak and A. Janusz, "Ensembles of Bireducts: Towards Robust Classification and Simple Representation," in *Proc. of FGIT 2011*, ser. LNCS, vol. 7105, 2011, pp. 64–77.
- [10] S. Widz and D. Ślęzak, "Rough Set Based Decision Support – Models Easy to Interpret," in *Selected Methods and Applications of Rough Sets in Management and Engineering*, ser. Advanced Information and Knowledge Processing, G. Peters, P. Lingras, D. Ślęzak, and Y. Yao, Eds. Springer, 2012, pp. 95–112.
- [11] M. J. Moshkov, M. Piliszczuk, and B. Zielosko, *Partial covers, reducts and decision rules in rough sets: theory and applications*. Springer, 2009, vol. 145.
- [12] W. Ziarko, "Probabilistic approach to rough sets," *International Journal of Approximate Reasoning*, vol. 49, no. 2, pp. 272–284, 2008.
- [13] G. Peters, P. Lingras, D. Ślęzak, and Y. Yao, *Rough sets: Selected methods and applications in management and engineering*. Springer Science & Business Media, 2012.
- [14] J. Wroblewski, "Ensembles of classifiers based on approximate reducts," *Fundam. Inform.*, vol. 47, no. 3-4, pp. 351–360, 2001. [Online]. Available: <http://content.iospress.com/articles/fundamenta-informaticae/f47-3-4-14>
- [15] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [16] D. Ślęzak and S. Widz, "Is It Important Which Rough-Set-Based Classifier Extraction and Voting Criteria Are Applied Together?" in *Proc. of Int. Conf. on Rough Sets and Current Trends in Computing (RSCTC)*, ser. LNAI, vol. 6086. Springer, 2010, pp. 187–196.